WHAT IS CLAIMED IS:

1. A method comprising:

generating a next virtual fetch address corresponding to an instruction fetch request;

determining whether a current physical address translation is valid for said next virtual fetch address in response to said generating, wherein said determining comprises detecting a change in the virtual page number of said next virtual fetch address relative to a virtual page number of a current virtual fetch address;

activating an ITLB circuit in response to determining that said current physical address translation is not valid for said next virtual fetch address; and

performing said instruction fetch using said current physical address translation without activating said ITLB circuit in response to determining that said current physical address translation is valid for said next virtual fetch address.

2. The method as recited in claim 1, wherein said determining that said current physical address translation is not valid for said next virtual fetch address further comprises detecting an ITLB update operation.

3. The method as recited in claim 2, wherein said detecting an ITLB update operation further comprises determining that an ITLB entry being modified corresponds to said current physical address translation.

4. The method as recited in claim 1, wherein said generating said next virtual fetch address comprises incrementing said current virtual fetch address by a fetch increment value, and wherein detecting a change in the virtual page number comprises detecting an arithmetic carry out of said incrementing at a bit position corresponding to a current virtual page size.

5. The method as recited in claim 1, wherein said generating said next virtual fetch address comprises selecting a branch target address, and wherein said detecting a change in the virtual page number comprises detecting a difference between a virtual page number of said branch target address and a virtual page number of said current virtual fetch address.

6. The method as recited in claim 1, wherein said generating said next virtual fetch address comprises selecting a branch target address, and wherein said detecting a change in the virtual page number comprises receiving branch prediction information indicative of a virtual page transition.

7. The method as recited in claim 1, further comprising buffering one or more physical address translations, and wherein said determining whether a current physical address translation is valid for said next virtual fetch address further comprises:

determining whether a given buffered physical address translation is valid for said next virtual fetch address; and

selecting said given buffered physical address translation as said current physical address translation in response to determining that said given buffered physical address translation is valid for said next virtual fetch address.

8. A fetch address generator comprising:

incrementor logic configured to generate a next virtual fetch address corresponding to an instruction fetch request; and

ITLB filtering logic coupled to said incrementor logic, wherein said ITLB filtering logic is configured to:

determine whether a current physical address translation is valid for said next virtual fetch address in response to said generating, wherein said determining comprises detecting a change in the virtual page number of said next virtual fetch address relative to a virtual page number of a current virtual fetch address; and

activate an ITLB circuit in response to determining that said current physical address translation is not valid for said next virtual fetch address;

wherein said fetch address generator is configured to cause said instruction fetch to be performed using said current physical address translation without activating said ITLB circuit in response to said ITLB filtering logic determining that said current physical address translation is valid for said next virtual fetch address.

9. The fetch address generator as recited in claim 8, wherein said determining that said current physical address translation is not valid for said next virtual fetch address further comprises detecting an ITLB update operation.

10. The fetch address generator as recited in claim 9, wherein said detecting an ITLB update operation further comprises determining that an ITLB entry being modified corresponds to said current physical address translation.

11. The fetch address generator as recited in claim 8, wherein said incrementor logic is further configured to increment said current virtual fetch address by a fetch increment value, and wherein detecting a change in the virtual page number comprises detecting an arithmetic carry out of said incrementor logic at a bit position corresponding to a current virtual page size.

12. The fetch address generator as recited in claim 8, wherein said incrementor logic is further configured to select a branch target address, and wherein said detecting a change in the virtual page number comprises detecting a difference between a virtual page number of said branch target address and a virtual page number of said current virtual fetch address.

13. The fetch address generator as recited in claim 8, wherein said incrementor logic is further configured to select a branch target address, and wherein said detecting a change in the virtual page number comprises receiving branch prediction information indicative of a virtual page transition.

14. The fetch address generator as recited in claim 8, wherein said ITLB filtering logic is further configured to:

buffer one or more physical address translations;

determine whether a given buffered physical address translation is valid for said next virtual fetch address; and

select said given buffered physical address translation as said current physical address translation in response to determining that said given buffered physical address translation is valid for said next virtual fetch address.

15. A microprocessor comprising:

an execution unit configured to execute instructions; and

instruction fetch logic coupled to said execution unit and configured to:

generate a next virtual fetch address corresponding to an instruction fetch request;

determine whether a current physical address translation is valid for said next virtual fetch address in response to said generating, wherein said determining comprises detecting a change in the virtual page number of said next virtual fetch address relative to a virtual page number of a current virtual fetch address;

activate an ITLB circuit in response to determining that said current physical address translation is not valid for said next virtual fetch address; and

perform said instruction fetch using said current physical address translation without activating said ITLB circuit in response to determining that said current physical address translation is valid for said next virtual fetch address.

16. The microprocessor as recited in claim 15, wherein said determining that said current physical address translation is not valid for said next virtual fetch address further comprises detecting an ITLB update operation.

17. The microprocessor as recited in claim 16, wherein said detecting an ITLB update operation further comprises determining that an ITLB entry being modified corresponds to said current physical address translation.

18. The microprocessor as recited in claim 15, wherein said generating said next virtual fetch address comprises incrementing said current virtual fetch address by a fetch increment value, and wherein detecting a change in the virtual page number comprises detecting an arithmetic carry out of said incrementing at a bit position corresponding to a current virtual page size.

19. The microprocessor as recited in claim 15, wherein said generating said next virtual fetch address comprises selecting a branch target address, and wherein said detecting a change in the virtual page number comprises detecting a difference between a virtual page number of said branch target address and a virtual page number of said current virtual fetch address.

20. The microprocessor as recited in claim 15, wherein said generating said next virtual fetch address comprises selecting a branch target address, and wherein said

detecting a change in the virtual page number comprises receiving branch prediction information indicative of a virtual page transition.

21. The microprocessor as recited in claim 15, wherein said instruction fetch logic is further configured to buffer one or more physical address translations, and wherein said determining whether a current physical address translation is valid for said next virtual fetch address further comprises:

determining whether a given buffered physical address translation is valid for said next virtual fetch address; and

selecting said given buffered physical address translation as said current physical address translation in response to determining that said given buffered physical address translation is valid for said next virtual fetch address.